What is claimed is:

1   1.   A database system comprising:

2   a storage to store a view containing results of a cube-based operation on at least

3   one base table, the view containing a first result set for a group-by on a first grouping set,

4   and a second result set for a group-by on a second grouping set; and

5   a controller, in response to a change to the at least one base table, to:

6   update the first result set by computing a change to the first result set

7   based on a change in the at least one base table; and

8   update the second result set by computing a change to the second result set

9   based on the change to the first result set.

1   2.   The database system of claim 1, wherein the first grouping set has a first number

2   of grouping attributes, and the second grouping set has a second number of grouping

3   attributes, the first number being greater than the second number.

1   3.   The database system of claim 2, wherein the view further contains a third result

2   set for a group-by on a third grouping set having a third number of grouping attributes,

3   the third number less than the second number,

4   the controller to further update the third result set by computing a change to the

5   third result set based on the change to the second result set.

1   4.   The database system of claim 1, wherein the view contains results of a group-by

2   cube operation.

1   5.   The database system of claim 1, wherein the view contains results of a group-by

2   partial cube operation.

17

1 6.   The database system of claim 1, further comprising plural access modules and
2 plural storage modules, the access modules to enable parallel access of data in the plural
3 storage modules.

1 7.   The database system of claim 6, wherein the controller is adapted to distribute
2 rows in the first result set across the access modules based on a hash of columns of the
3 second grouping set and at least another column that is assigned a predefined value.

1 8.   The database system of claim 7, wherein the view contains results of a cube
2 operation specified by a cube function on plural columns, the at least another column
3 being one of the plural columns of the cube function that is not in the second grouping
4 set.

1 9.   The database system of claim 7, wherein the view is distributed across the access
2 modules such that plural portions of the view reside in respective storage modules, and
3 wherein the rows in the first result set are distributed across the access modules according
4 to the hash to enable:
5   each access module to locally perform a merge and aggregate operation on the
6 rows of the first result set to produce rows of the second result set; and
7   each access module to locally merge the rows of the second result set into a
8 respective portion of the view without having to first redistribute the rows of the second
9 result set.

1 10.   The database system of claim 1, wherein the controller is adapted to further:
2   receive a query specifying a group-by operation; and
3   determine whether an answer for the query specifying the group-by operation can
4 be satisfied from the view.

1   11.   The database system of claim 10, wherein the query specifies a group-by

2   operation on grouping sets S, and the view contains result sets for grouping sets C,

3   the controller to determine whether S is a subset of C to determine whether the

4   answer for the query can be satisfied from the view.

1

1   12.   The database system of claim 11, wherein the controller is adapted to modify a

2   WHERE clause of the query in response to determining that S is a subset of C.

1

1   13.   A method for use in a database system, comprising: ⌐⁄

2   storing a view containing results of a cube-based operation on at least one base

3   table, the view containing result sets for group-bys on respective grouping sets;

4   updating a first result set by computing a change to the first result set based on a

5   change in the at least one base table; and

6   updating a second result set by computing a change to the second result set based

7   on the change to the first result set.

1   14.   The method of claim 13, wherein updating the first result set comprises updating

2   the first result set for the group-by on a first grouping set that has a greater number of

3   columns than a second grouping set corresponding to the second result set.

1   15.   The method of claim 13, further comprising updating a third result set by

2   computing a change to the third result set based on the change to the second result set.

1   16.   The method of claim 15, further comprising updating a fourth result set by

2   computing a change to the fourth result set based on the change to the third result set.

1   17.   The method of claim 13, wherein the database system has plural storage modules

2   to store respective portions of the view, and plural access modules to manage access of

3   respective storage modules,

4   wherein updating the first result set and second result set are performed in parallel

5   by the plural access modules.

1   18.   The method of claim 17, further comprising distributing rows of the first and

2   second result sets across the plural access modules.

1   19.   The method of claim 18, wherein the first result set corresponds to a group-by on

2   a first grouping set having N columns, and the second result set corresponds to a group-

3   by on a second grouping set having N-1 columns, and wherein distributing the first result

4   set to compute the second result set comprises distributing the first result set based on a

5   hash of the N columns, with the column in the first grouping set not present in the second

6   grouping set assigned a predefined value.

1   20.   The method of claim 19, further comprising:

2   updating a third result set by computing a change to the third result set based on

3   the change to the second result set, wherein the third result set corresponds to a group-by

4   on a third grouping set having N-2 columns,

5   wherein distributing the second result set across the access modules to compute

6   the third result set is based on a hash of the N columns, with the columns in the first

7   grouping set not appearing in the third grouping set each assigned to the predefined

8   value.

1   21.   The method of claim 20, wherein storing the view comprises storing a view for a

2   cube operation based on a cube function of the N columns.

1   22.   An article comprising at least one storage medium containing instructions that

2   when executed cause a database system to:

3   store a view containing results of a cube-based operation on at least one base

4   table, the view containing result sets for group-bys on respective grouping sets;

5   update a first result set by computing a change to the first result set based on a

6   change in the at least one base table; and

7   update a second result set by computing a change to the second result set based on

8   the change to the first result set.

1   23.   The article of claim 22, wherein updating the first result set comprises updating

2   the first result set for the group-by on a first grouping set that has a greater number of

3   columns than a second grouping set corresponding to the second result set.

1   24.   The article of claim 22, wherein the instructions when executed cause the

2   database system to further update a third result set by computing a change to the third

3   result set based on the change to the second result set.

1   25.   The article of claim 22, wherein the database system has plural storage modules to

2   store respective portions of the view, and plural access modules to manage access of

3   respective storage modules,

4   wherein updating the first result set and second result set are performed in parallel

5   by the plural access modules.

1   26.   The article of claim 25, wherein the instructions when executed cause the

2   database system to further distribute rows of the first and second result sets across the

3   plural access modules.

1   27.   The article of claim 26, wherein the first result set corresponds to a group-by on a

2   first grouping set having N columns, and the second result set corresponds to a group-by

3   on a second grouping set having N-1 columns, and wherein distributing the first result set

4   to compute the second result set comprises distributing the first result set based on a hash

5   of the N columns, with the column in the first grouping set not present in the second

6   grouping set assigned a predefined value.

1   28.   The article of claim 27, wherein storing the view comprises storing a view for a

2   cube operation based on a cube function of the N columns.